REMARKS

In response to the Official Action mailed February 13, 2004, Applicant requests reconsideration. In this Response, no claims are added, canceled, or amended, so that claims 67-76 remain pending. No new matter has been added.

The Official Action rejected claims 67-76 as unpatentable over Dyer et al. (US Patent 5,754,849, hereinafter Dyer) in view of "*Microsoft Message Queue Server Reviewer's Guide*" (hereinafter "the MMQS Guide") and Vanderbilt et al. (US Patent 5,793,965, hereinafter Vanderbilt). That rejection is respectfully traversed.

As to claims 67 and 72, the Official Action asserted that Dyer teaches all the limitations except (1) that the first/second message communication machines are the first/ second message queuing machines; and (2) steps of the recipient application identifying a data element in the unserialized dictionary object received from the second queuing machine having a data type not recognized by the recipient application, and the recipient application sending a query to the first message queuing machine to learn about said data type. As to (1), the Official Action asserted that it would have been obvious to implement the first/second message communication machines of Dyer as respective first/second message queuing machines in view of the teachings of the MMQS Guide.

Applicant respectfully submits that the rejection is not fully supported because the combined references do not teach all the claim limitations. Specifically, as to (2), contrary to the assertions of the Official Action, the Vanderbilt reference does not teach or suggest a recipient application "identifying a data element in the unserialized dictionary object received from the second message queuing machine having a data type not recognized by the recipient application" and "sending a query to the first message queuing machine to learn about said data type." The Official Action argues that:

> "Vanderbilt teaches the sender and receiver computers, ie, the first and the second computers, can be the same computer (col., 17, lines 27-38; col. 18, lines 16-17, 63-64). In other words, the IS_A function in this situation is used to learn the type information about a local object. Dyer teaches such local objects includes those received from the remote machine / second queuing machine. Therefore, the combination of Dyer and Vanderbilt would have provided using the IS_A function to learn about the type information about a local object which is received from a remote machine / second queuing machine."

However, neither Dyer nor Vanderbilt discloses receiving an unrecognized object. In Dyer, the objects are all recognized by their type indicators (see Abstract of Dyer). In Vanderbilt, no objects are passed, so no object can be received. It does not matter whether these objects are located on the same or on remote computers. The objects of Vanderbilt merely execute functions and pass messages between each other – the objects themselves are not included in messages passed by applications (see column 6, lines 12-15 and lines 52-59 of Vanderbilt). Accordingly, there is no object to be received by a recipient application, and thus no object to be unrecognized.

Furthermore, neither Dyer nor Vanderbilt teaches sending a query to the first message queuing machine to learn about the data type. The Official Action appears to erroneously construe the phrase "learn about the data type" as "learn what the data type is." In Vanderbilt, the IS_A function is a boolean function, and returns only true or false as to whether the object is of a type specified by the query (see column 10, lines 2-15 of Vanderbilt). Thus, Vanderbilt discloses merely that the IS_A function determines whether the object is of a type already known to the querying client; the querying client does not "learn about the data type" in this way. The client may discover what the data type is, but in this way the client is learning about the object, not about the data type. In fact, if the client in Vanderbilt were unfamiliar with the data type of the object, the client could never discover it because IS_A returns only true or false (see column 10, lines 2-15 of Vanderbilt). Accordingly, the combination of Dyer and Vanderbilt does not teach "learning about the data type" of an unrecognized object.

Moreover, there is no motivation to combine Dyer and Vanderbilt as suggested by the Official Action because the proposed modification would not work. In Dyer, objects are identified using type codes (see Abstract of Dyer). In Vanderbilt, a client uses an IS_A function to ask an object whether it is a specified type, and receives a yes or no answer (see column 10, lines 2-15 of Vanderbilt). If an application in Dyer did not recognize the type of an object, it could not use the IS_A function of Vanderbilt to learn about the unrecognized type. The application could only ask the sender a series of yes or no questions (the IS_A function) regarding the already known types. If the type is not known by the application, there is no way to use the IS_A function to learn more about the type, such as how to handle or process the object. Accordingly, the combination of Dyer and Vanderbilt would not work to teach or suggest the present invention.

As to claims 68 and 73, the combination of Dyer and Vanderbilt further fails to teach or suggest sending the query to the sending application. Neither Dyer nor Vanderbilt discloses a

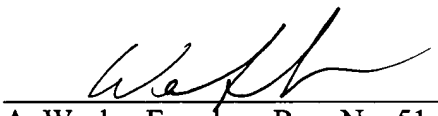
sending application that sends a message containing an unrecognized object, as previously stated. In Vanderbilt, the query is sent to a proxy object or the Object Request Broker, neither of which can be construed as a sending application (see column 10, line 25 to column 11, line 40 of Vanderbilt). In fact, the client application sending the IS_A query is the sending application in Vanderbilt.

As to claims 71 and 76, the Official Action does not point out how the combination of Dyer and Vanderbilt teaches or suggests "determining, by a message queuing server of the first message queuing machine, whether the message object supports persistence." In fact, the combination does not. Neither Dyer nor Vanderbilt makes any mention of determining whether a message object supports persistence.

Because the combination of Dyer and Vanderbilt fails to teach or suggest all of the limitations of the pending claims, and because there is no motivation to combine Dyer and Vanderbilt, *prima facie* obviousness has not been established. Accordingly, the rejection of claims 67-76 is erroneous and should be withdrawn.

Reconsideration and withdrawal of the rejection is earnestly solicited.

Respectfully submitted,



A. Wesley Ferrebee, Reg. No. 51,312
LEYDIG, VOIT & MAYER
700 Thirteenth Street, N.W., Suite 300
Washington, DC 20005-3960
(202) 737-6770 (telephone)
(202) 737-6776 (facsimile)

Date: 5/13/04